Contents lists available at SciVerse ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Short communication

Enabling leg-based guidance on top of waypoint-based autopilots for UAS

Xavier Prats [*], Eduard Santamaria, Luis Delgado, Noel Trillo, Enric Pastor

Technical University of Catalonia (UPC), Castelldefels School of Telecommunications and Aerospace Engineering, Esteve Terradas, 7, 08860 Castelldefels, Spain

ARTICLE INFO

Article history:
Received 20 May 2011
Received in revised form 6 September 2011
Accepted 20 September 2011
Available online 24 September 2011

Keywords:
Leg-based guidance
RNAV navigation
Autopilot
UAS

ABSTRACT

This paper presents a methodology to extend the guidance functionalities of Commercial Off-The-Shelf autopilots currently available for Unmanned Aircraft Systems (UAS). Providing that most autopilots only support elemental waypoint-based guidance, this technique allows the aircraft to follow leg-based flight plans without needing to modify the internal control algorithms of the autopilot. It is discussed how to provide Direct to Fix, Track to Fix and Hold to Fix path terminators (along with Fly-Over and Fly-By waypoints) to basic autopilots able to natively execute only a limited set of legs. Preliminary results show the feasibility of the proposal with flight simulations that used a flexible and reconfigurable UAS architecture specifically designed to avoid dependencies with a single or particular autopilot solution.

1. Introduction

Unmanned Aircraft Systems (UAS) are, at present, mainly designed for military missions and very few civil applications have been developed so far because of the lack of an appropriate regulation framework concerning their certification, airworthiness and operations. An important consequence is that the development of UAS has always been highly dependent on the mission and the flight scenario. Thus, very specific and non-flexible systems exist nowadays to control the desired flight plan, the UAS mission, the sensor activation/configuration, the data storage, etc. This fact may delay and increase the implementation costs (and risk) of a new UAS application. The core of a UAS is probably the autopilot and, in order to develop all necessary mission avionics on top of it, it is essential to understand how this device works, what kind of inputs supports and what are its capabilities.

There are a considerable amount of Commercial Off-The-Shelf (COTS) autopilots designed for UAS (refer for instance to the survey done in [2]). A well-known drawback regarding this technology is that the flight plan definition is composed by just a collection of statically defined waypoints (or hand-manipulated by the UAS operator), which are flown in sequence. Using a basic list of waypoints as the mechanism for flight planning specification has several important limitations, such as the difficulty to specify complex trajectories (needed for most of the UAS typical missions), the lack of support for specifying repetitive or conditional behaviour in the flight plan, etc. To address these limitations, in [8,6] a new flight plan specification language has been proposed that uses higher level constructs, with richer semantics, and which enables the UAS flight to be adapted to the circumstances encountered during the mission.

Besides basic trajectories, the flight plan specification supports constructs for conditional and iterative behaviour together with mission-oriented area scanning and other built-in patterns. The flight plan submitted to the UAS contains a nominal path that can be modified in real-time by updating a number of specific parameters. Moreover, the flight plan can be accompanied by alternatives to respond to emergency situations as well as specific procedures for approach and departure operations. The details of this flight plan specification language are found in [8], which also describes the overall architecture of the system with a special emphasis on the software components involved in the execution of the flight plan. These software components run on embedded hardware installed on-board the UAS. Additionally, the human machine interfaces required to fully exploit the new dynamic characteristics offered by the flight plan language are detailed in [6]. The aforementioned references, however, do not go into detail regarding the translation mechanism used to generate waypoints for the different flight plan primitives. This article provides insights on how this translation process takes place and shows how the proposed methodology can be used to implement primitives found in the flight plan specification language on top of autopilots with different navigation capabilities.

Current COTS autopilots for UAS do not support any of these capabilities and the required guidance functionalities. Based on Area

* Corresponding author.
E-mail addresses: xavier.prats@upc.edu (X. Prats), santama@ac.upc.edu (E. Santamaria), luis.delgado@upc.edu (L. Delgado), noel.trillo@estudiant.upc.edu (N. Trillo), enric@ac.upc.edu (E. Pastor).
URL: http://icarus.upc.edu (X. Prats).

1270-9638/$ – see front matter © 2011 Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.ast.2011.09.006

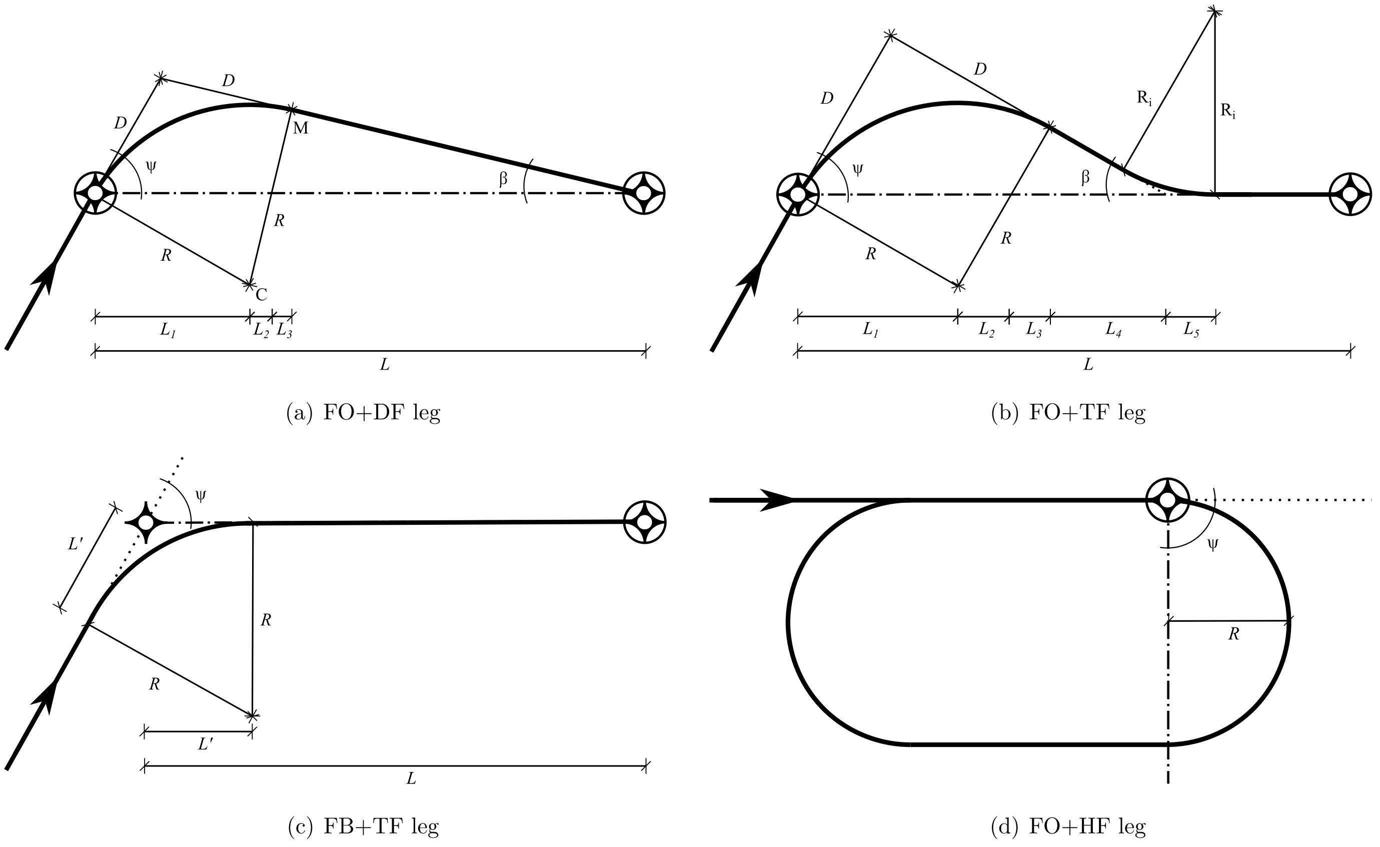


Fig. 1. Characteristic magnitudes of the four legs (waypoint+path terminator) considered in this paper.

Navigation (RNAV),[1] our flight plan specification language proposes the *leg* as main flight planning component. A leg determines not only a destination point, but also the trajectory that the aircraft should follow in order to reach it. Since most UAS autopilots only provide elemental waypoint-based guidance, a method for translating flight plan legs to a sequence of waypoints which can be correctly interpreted by the autopilot is needed. This paper focuses on the computations required to transform RNAV leg-based guidance primitives into sequences of "elemental waypoints", and how the proposed methodology can be used to implement primitives found in the flight plan specification language.

It should be noted that this paper is not proposing a new path planning technique, used for example in highly constrained environments (see [5]); but a way to enhance existing COTS autopilots with an extra set of RNAV legs. Thence, the safe and appropriate design of the flight plan is out of the scope of this paper, since it is assumed to be compliant with RNAV standards and current regulations regarding aircraft procedure design criteria [4]. Thus, the proposed methodology creates a list of autopilot-compatible waypoints by erasing the original ones in the flight plan and/or by adding some extra *synthetic* waypoints which can be executed by the considered autopilot. Since commercial autopilots are offered as closed systems, this strategy allows to extend the autopilot capabilities without needing to modify its internal control algorithms. In particular, this paper describes how to provide new RNAV leg-based functionalities to three hypothetical COTS systems which are able to execute only one type of waypoint and leg. Section 2 of this paper presents the implemented functionalities while some preliminary simulation results are shown in Section 3.

[1] A method of navigation that allows to fly routes that can be defined between arbitrary waypoints within the coverage of a network of navigation beacons or within the limits of a self contained positioning system, or a combination of both.

2. Implemented functionalities

In manned aviation, RNAV procedures have been progressively introduced in the last two decades, along with Flight Management Systems (FMS) on-board the aircraft. In order to translate RNAV leg-based procedures into a suitable code for these systems, the industry has developed the *Path and Termination* concept, along with a database standard published by Aeronautical Radio, Inc. (ARINC) [1]. This standard details how databases for FMS are to be coded. Path terminators provide the means to translate the text and the routes depicted on the navigation plates used by pilots into FMS readable code.

In short, a Path terminator defines a specific flight path and a specific type of termination for each leg. Leg types are identified by a two letter code that describes the path (e.g., heading, course, track, etc.) and the termination condition (e.g., the path terminates at a fix, an altitude, a given distance to a radiobeacon, a given time, etc.). Besides path terminators, two basic waypoint types exist in RNAV: Fly-Over (FO) and Fly-By (FB) waypoints. The former entail the actual over-fly of the waypoint, before heading to the next waypoint. On the other hand, in an FB waypoint the aircraft starts turning before reaching the waypoint in such a way that it is obtained a turning arc tangential with the two flight segments that precede and follow the waypoint.

For example, Figs. 1(a) and 1(b) display two different path terminators following a Fly-Over waypoint. In Fig. 1(a) a *Direct to Fix* (DF) path terminator is executed, meaning that the aircraft will head direct to the next waypoint after completing the turn, regardless of its actual position. On the other hand, Fig. 1(b) displays a *Track to Fix* (TF) path terminator where, after over-flying the turning waypoint, the aircraft joins the track existing between the over-flown waypoint and the next one. In Fig. 1(c) a TF path ter-

Table 1
Required modifications on the nominal flight plan in order to enhance a system capable of executing only one type of leg (waypoint+path terminator).

System capability	Implemented leg	Delete original WP?	Extra *virtual* WPs	Coordinates of the extra waypoints (WPs)	
FO+DF	FO+TF	No	WP1	$x_1 = L_1 + L_2 + L_3 + L_4 - L_5 \cos\beta$	$y_1 = L_5 \sin\beta$
	FB+TF	Yes	WP1	$x_1 = -L' \cdot \cos\psi$	$y_1 = -L' \cdot \sin\psi$
	FO+HF	No	WP1	$x_1 = -L''$	$y_1 = -2R$
FO+TF	FO+DF	No	WP1	$x_1 = L_1 + L_2 + L_3$	$y_1 = (L - L_1 - L_2 - L_3)\tan\beta$
	FB+TF	Yes	WP1	$x_1 = -L' \cdot \cos\psi$	$y_1 = -L' \cdot \sin\psi$
			WP2	$x_2 = L'$	$y_2 = 0$
	FO+HF	No	WP1	$x_1 = 0$	$y_1 = -2R$
			WP2	$x_2 = -L''$	$y_2 = -2R$
			WP3	$x_3 = -L''$	$y_3 = 0$
FB+TF	FO+DF	Yes	WP1	$x_1 = D\cos\psi$	$y_1 = D\sin\psi$
	FO+TF	Yes	WP1	$x_1 = D\cos\psi$	$y_1 = D\sin\psi$
			WP2	$x_2 = L_1 + L_2 + L_3 + L_4$	$y_2 = 0$
	FO+HF	Yes	WP1	$x_1 = R$	$y_1 = 0$
			WP2	$x_2 = R$	$y_2 = -2R$
			WP3	$x_3 = -(L'' + R)$	$y_3 = -2R$
			WP4	$x_4 = -(L'' + R)$	$y_4 = 0$

minator is shown after a Fly-By type waypoint, while in Fig. 1(d) a *Hold to Fix* (HF) follows the Fly-Over waypoint.

There exist up to 23 different path terminators, even if most of the FMS can usually execute a small set of them [3]. This paper describes how to provide new RNAV functionalities to three hypothetical systems which are able to execute only one type of leg (waypoint+path terminator). These three studied systems are:

- **FO+DF**: System capable of Fly-Over (FO) waypoints followed by Direct to Fix (DF) path terminators.
- **FO+TF**: System capable of Fly-Over (FO) waypoints followed by Track to Fix (TF) path terminators.
- **FB+TF**: System capable of Fly-By (FB) waypoints followed by Track to Fix (TF) path terminators.

Among all RNAV path terminators, DF, TF and HF are the most basic and common ones, allowing an autopilot to execute a wide range of basic RNAV procedures already published in many countries [3]. In this paper, the following legs will be implemented for each of the three previous considered systems: FO+DF, FO+TF, FB+TF and FO+HF.[2] In Fig. 1, the nominal trajectory and associated magnitudes of these four transitions, when considering a course change of angle ψ, are shown.

Assuming an aircraft performing a coordinated turn at a constant bank angle ϕ and with no altitude change, the nominal turn radius is given by:

$$R = \frac{v^2}{g\tan\phi} \tag{1}$$

where v is the aircraft airspeed, while $g \simeq 9.81\ \mathrm{m\,s^{-1}}$ is the acceleration of the gravity. The bank angle is an aircraft dependent parameter, taking in general, values around $\phi = 25°$ in manned aircraft [7]. For UAS, however, higher values can be foreseen.

For an FO+DF transition (Fig. 1(a)), the aircraft turns in such a way that its new heading leads directly to the next waypoint. Therefore, the interception angle for this waypoint is a function of the course change (ψ), the length of the leg (L) and the turn radius (R). The analytical computation of this angle $\beta(\psi, L, R)$ is given in Appendix A of this paper. Conversely, for an FO+TF transition (see Fig. 1(b)), this interception angle is fixed beforehand to a typical value of $\beta = 30°$ [4].

[2] Note that the combination FB+DF is senseless and that the HF path terminator must always follow an FO waypoint [1].

From Fig. 1 and after some basic trigonometric analysis, some characteristic length magnitudes describing each transition can be computed as follows:

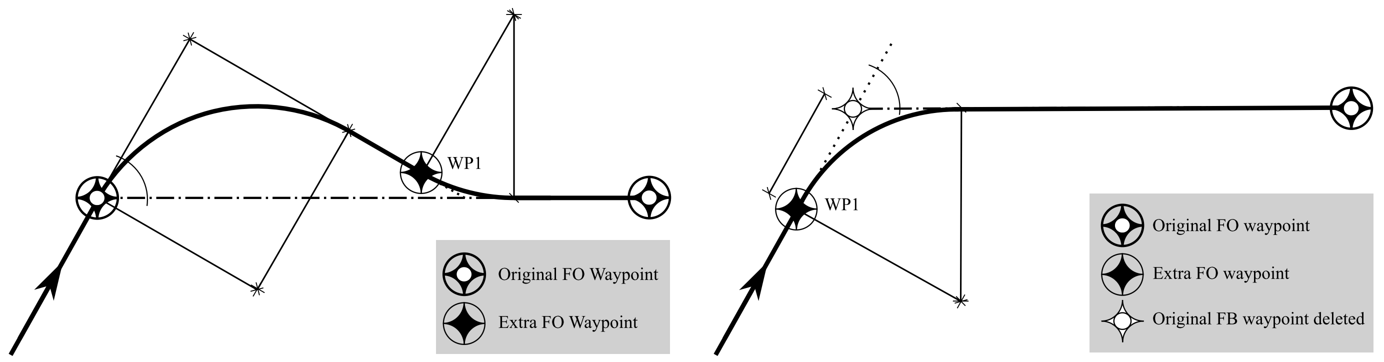
$$L_1 = R\sin\psi \qquad L_2 = R\cos\psi\tan\beta$$
$$L_3 = R\sin\beta\left(1 - \frac{\cos\psi}{\cos\beta}\right) \qquad L_4 = R\frac{\cos^2\beta}{\sin\beta}\left(1 - \frac{\cos\psi}{\cos\beta}\right)$$
$$L_5 = R_i\tan\frac{\beta}{2} \qquad L' = R\tan\frac{\psi}{2}$$
$$D = R\tan\left(\frac{\psi+\beta}{2}\right) \tag{2}$$

The methodology proposed in this paper to enhance the basic autopilot capabilities is by adding extra *virtual* waypoints and/or by eliminating the original waypoint to/from the original flight plan. This modification of the nominal list of waypoint is computed in a way that the trajectory obtained when executing the new flight plan with a basic autopilot, matches with the trajectory we would obtain had the original flight plan been executed with an autopilot capable to perform natively the required waypoint and path terminator types.

According to this methodology, Table 1 summarises all the modifications required on the nominal flight plan in order to virtually provide the autopilot with the four new leg capabilities (FO+TF, FO+DF, FB+TF and FO+HF). Thus, for each new implemented leg, the table shows the cases when the original waypoint should be deleted from the flight plan and how many extra *virtual* waypoints are needed to code. For each extra waypoint, its location is given in a Cartesian coordinate system (x, y), centred at the nominal waypoint of study and with the x axis aligned with the line joining this waypoint with the following waypoint in the flight plan sequence. In Figs. 2, 3 and 4, the location of these new waypoints is shown graphically for each of the new implemented leg. These figures correspond to autopilots capable to execute only FO+DF, FO+TF and FB+TF transitions, respectively.

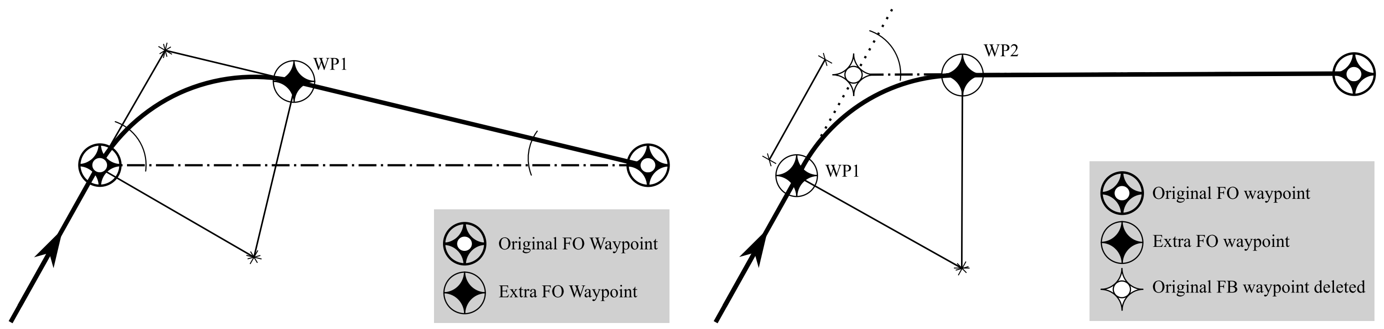
3. Preliminary implementation and simulation infrastructure

The implementation of the described methods forms part of an on-going effort by the authors to build a UAS architecture able to execute leg-based procedures that is not locked-in to a single autopilot solution. Fig. 5 displays the simulation environment that has been set up to test the presented approach. The Flight Plan Manager (FPMa) and the Virtual Autopilot System (VAS) are software components, on-board the UAS platform, that communicate



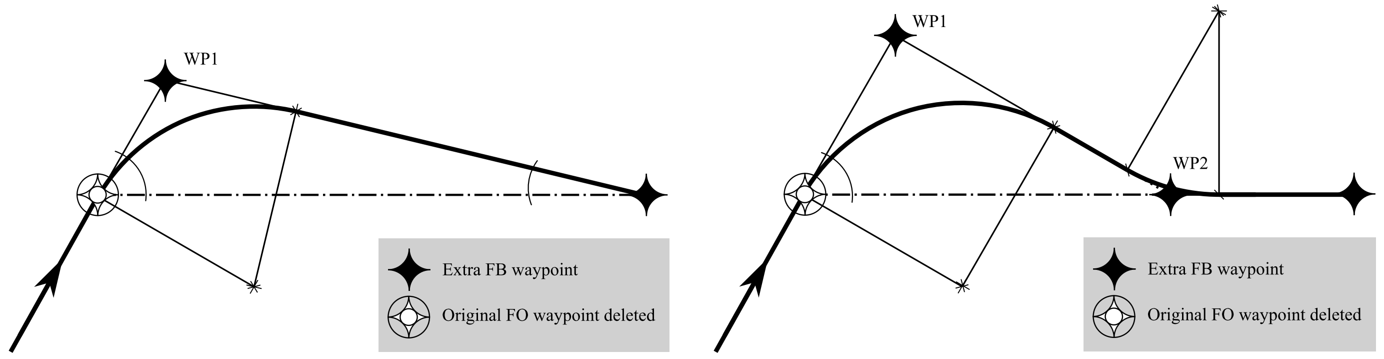
(a) Enabling a FO+TF leg (b) Enabling a FB+TF leg

Fig. 2. Enhancing an autopilot capable to execute only FO+DF legs.



(a) Enabling a FO+DF leg (b) Enabling a FB+TF leg

Fig. 3. Enhancing an autopilot capable to execute only FO+TF legs.



(a) Enabling a FO+DF leg (b) Enabling a FO+TF leg

Fig. 4. Enhancing an autopilot capable to execute only FB+TF legs.

with each other through a Local Area Network. A document describing the flight plan using our leg-based specification language is submitted to the FPMa to perform its execution. This component, by using the above presented equations, computes the sequence of waypoints that the flight plan translates into, and sends them to the VAS; which in turn, provides a hardware-independent interface to interact with the autopilot. As shown in the figure, the implementation of the VAS is divided into two blocks: one facing the network and another one facing the UAS platform. While the former is always the same, the latter depends on the actual autopilot system and implements the communication protocols required by the autopilot. Therefore, as illustrated, one could have an implementation that is able to operate with a given commercial autopilot and another one for interacting with a particular flight simulator. The installed autopilot system is ultimately responsible for implementing the guidance and control loops.

The presented simulations have been performed by using the FlightGear open source flight simulator.[3] Fig. 6(a) displays the results of an example simulation when the VAS interacts with an autopilot capable of executing only FO waypoints followed by DF path terminators. The figure displays the flight plan legs as straight lines connecting different waypoints, which can be either FB or FO. The actual list of waypoints generated by the FPMa is drawn on top, together with the actual flight trajectory obtained after the simulation. Leg A is a TF leg that connects an FO waypoint (2)

[3] See http://www.flightgear.org.

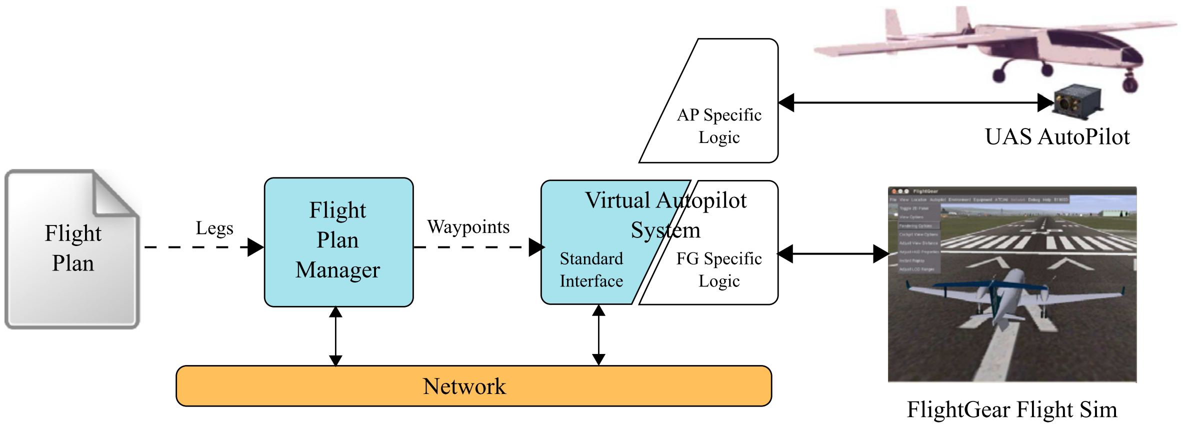
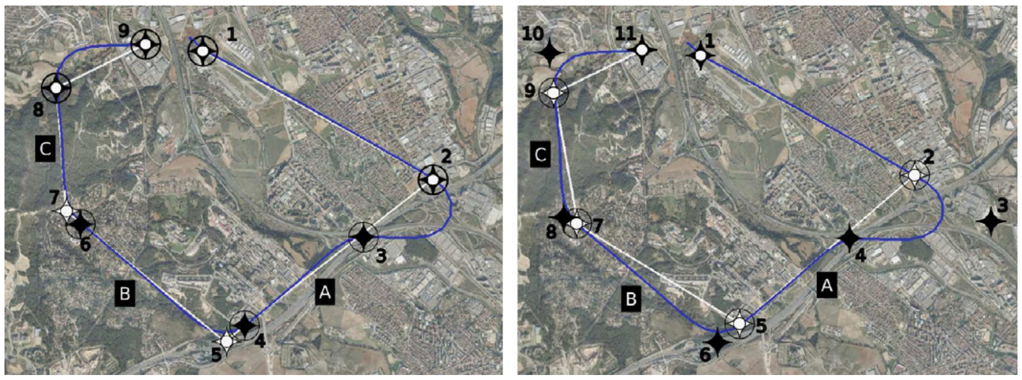


Fig. 5. Simulation environment.



(a) Enhancing an autopilot capable to execute only FO+DF legs

(b) Enhancing an autopilot capable to execute only FB+TF legs

Fig. 6. Simulation results using systems with different basic leg capabilities.

to an FB destination waypoint (5). Since the simulated autopilot deals natively with FO waypoints, no special treatment is required for waypoint (2). An extra waypoint (3) is needed to intercept the track and the original destination waypoint (5) is moved to a new position (4) to actually perform a fly-by manoeuvre. Leg B is also a TF leg and as in leg A, the destination waypoint (7) is moved to a new position (6) in order to perform a fly-by. With the destination waypoint (8) of leg C being an FO followed by a DF leg, no special waypoint transformations are required for the rest of the flight plan.

A different flight plan and its execution are shown in Fig. 6(b) to demonstrate how the system deals with an autopilot that supports only FB waypoints followed by TF path terminators. Leg A of this flight plan is also a TF leg that starts with an FO waypoint (2). Since FO waypoints are not natively supported, waypoint 2 needs to be removed and two extra FB waypoints are added (3 and 4). Legs B and C are DF legs that start with an FO waypoint (respectively, waypoints 5 and 7) and in both cases, the aircraft follows the track between a displaced initial waypoint (6 and 8) and the leg's destination.

4. Conclusion

Area Navigation, coupled with other technologies, is enabling a move towards trajectory-based operations. Bringing trajectory-based capabilities to UAS will make them easier to integrate into non-segregated airspace and also increase their ability to perform missions which require complex trajectories to be flown (such as loitering over certain areas). Most of current commercially available autopilot solutions, however, only provide waypoint-based guidance. Moreover, due to the closed nature of these systems, it is not possible to modify them in order to implement leg-based guidance. A possible solution to enhance such basic autopilots is to conveniently modify the original flight plan, as it has been shown in this paper.

Since the presented simulations have been carried out without considering wind conditions, arguably, the results only confirm the correctness of the analytical equations proposed and their implementation. Nevertheless, the simulation infrastructure provides a valuable test-bed for further developments and work is underway to include wind estimations into the presented waypoint generation equations. Moreover, a sensitivity analysis of the uncertainties in parameters such as aircraft airspeed and bank angles, is also foreseen. Besides that, we also plan to close the loop between the Flight Plan Manager and the Virtual Autopilot so that the former will be able to keep track of the deviations from the nominal path and recompute waypoints to adjust the trajectory.

Acknowledgements

This work has been partially funded by Ministry of Science and Education of Spain under contract CICYT TIN 2010-18989. This work has been also co-financed by the European Organisation for the Safety of Air Navigation (EUROCONTROL) under its CARE INO III program. The content of the work does not necessarily reflect the official position of EUROCONTROL on the matter.

Appendix A. Interception angle computation

For a generic FO+DF leg, the interception angle β can be computed analytically after finding the coordinates of point M, as shown in Fig. 1(a). At this point, the arc describing the turning trajectory that follows after the FO waypoint is tangential to the flight segment that directly leads to the next waypoint. Let (x, y) be the coordinates of this point in a Cartesian coordinate system centred, this time, at the centre of the arc (point C). Let (x_p, y_p) be the coordinates of the following waypoint in the flight plan list. Then, point M coordinates can be found by solving the following two equation system:

$$\begin{cases} x^2+y^2=R^2 \\ \dfrac{y}{x}\cdot\dfrac{y-y_p}{x-x_p}=-1 \end{cases} \tag{3}$$

First equation imposes that point M belongs to the circumference centred at C, while second equation forces the tangency condition with the line that joins point M with the following waypoint. Two solutions are found for this equation system, being one of them the coordinates for point M:

$$\begin{aligned} x &= -\frac{R}{x_p}\left(-R+y_p\frac{Ry_p+\sqrt{-x_p^2R^2+x_p^4+y_p^2x_p^2}}{x_p^2+y_p^2}\right) \\ y &= Ry_p\left(\frac{Ry_p+\sqrt{-x_p^2R^2+x_p^4+y_p^2x_p^2}}{x_p^2+y_p^2}\right) \end{aligned} \tag{4}$$

Furthermore, the interception angle is given by:

$$\beta=\arctan\left(\frac{y-y_p}{x_p-x}\right) \tag{5}$$

and in turn, (x_p, y_p) coordinates can be expressed as:

$$x_p=L-R\sin\psi \qquad y_p=R\cos\psi \tag{6}$$

Then, by using Eqs. (4), (5) and (6), the interception angle can be expressed in function of the course change angle ψ, the leg length (L) and the turn radius (R) as:

$$\begin{aligned} \beta(\psi, L, R) = \arctan\Bigl(-R \\ \times\frac{L^3\mathrm{c}(\psi)-\frac{3}{2}RL^2\mathrm{s}(2\psi)-2R^2L\mathrm{c}^3(\psi)+2R^2L\mathrm{c}(\psi)-LA+RA\mathrm{s}(\psi)}{L^4-4RL^3\mathrm{s}(\psi)-5R^2L^2\mathrm{c}^2(\psi)+5R^2L^2+2R^3L\mathrm{s}(\psi)\mathrm{c}^2(\psi)-2R^3L\mathrm{s}(\psi)+R^2A\mathrm{c}(\psi)}\Bigr) \end{aligned} \tag{7}$$

where for the sake of compactness $\mathrm{s}(\cdot)$ and $\mathrm{c}(\cdot)$ correspond to $\sin(\cdot)$ and $\cos(\cdot)$ respectively, while:

$$\begin{aligned} A = \sqrt{L}\bigl(2R^3\mathrm{s}(\psi)\mathrm{c}^2(\psi)-2R^3\mathrm{s}(\psi) \\ -4RL^2\mathrm{s}(\psi)+L^3-5R^2L\mathrm{c}^2(\psi)+5R^2L\bigr)^{1/2} \end{aligned} \tag{8}$$

References

[1] ARINC, Navigation System Database, ARINC specification 424, 15th edition, Aeronautical Radio Inc., Annapolis, Maryland, USA, Feb. 2000.

[2] H. Chao, Y. Cao, Y. Chen, Autopilots for small fixed-wing unmanned air vehicles: A survey, in: International Conference on Mechatronics and Automation (ICMA), IEEE, Harbin, China, 2007, pp. 3144–3149, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4304064&isnumber=4303488.

[3] Eurocontrol, Guidance Material for the Design of Terminal Procedures for Area Navigation (DME/DME, B-GNSS, Baro-VNAV and RNP-RNAV), 3rd edition, Eurocontrol, Mar. 2003.

[4] ICAO, Procedures for Air Navigation Services – Aircraft Operations (PANS–OPS) – Volume II, Construction of Visual and Instrument Flight Procedures, 5th edition, International Civil Aviation Organisation, Montreal, Canada, doc. 8168, 2006.

[5] Y. Kuwata, J.P. How, Stable trajectory design for highly constrained environments using receding horizon control, in: Proceedings of the American Control Conference, AACC, 2004, pp. 902–907.

[6] E. Pastor, E. Santamaria, P. Royo, J. López, C. Barrado, On the design of a UAV flight plan monitoring and edition system, in: Proceedings of the IEEE Aerospace Conference, AIAA/IEEE, Big Sky, Montana, USA, 2010.

[7] R.W. Pratt (Ed.), Flight Control Systems: Practical Issues in Design and Implementation, IEE Control Engineering Series, Institution of Engineering and Technology, 2000.

[8] E. Santamaria, P. Royo, C. Barrado, E. Pastor, J. López, X. Prats, Mission aware flight planning for unmanned aerial systems, in: Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA, Honolulu, Hawaii, 2008, http://www.eurocontrol.int/eec/gallery/content/public/documents/Innovative_Studies/grants/2008/Mission_aware_flight_planning_for_UAVs.pdf.